

FUZZY-BASED PATH ANALYSIS

Imran Moez Khan¹, Yusof Zaw Zaw¹, Othman O. Khalifa¹ and Weng Kin Lai²

¹Department of Electrical and Computer Engineering,
Faculty of Engineering, International Islamic University Malaysia,
53100 Kuala Lumpur, Malaysia

²Centre for Multimodal Signal Processing, MIMOS Berhad
Technology Park Malaysia,
57000 Kuala Lumpur, Malaysia

ABSTRACT

Video surveillance can be a very powerful tool in the fight against crime, by accurately monitoring human activities. Nevertheless, most surveillance systems today provide only a passive form of site monitoring. Extensive video records may be kept to help find the instigator of criminal activities after the crime has been committed but preventive measures usually require human involvement. In addition to this, there is a need for large amounts of data storage to keep up to several terabytes of video streams that may be needed for later analysis. In order to achieve any form of real-time monitoring, guards often need to be employed to watch video feeds for hours on end to recognize suspicious, dangerous or potentially harmful situations. In a multi-camera scene monitoring system, this can be quite infeasible as there can be up to 20 to 50 cameras on average in a large building complex such as an airport or shopping malls. Intelligent video surveillance aims to reduce or even eliminate the need for human supervision of video feeds, and continuous recording. Having such a system will provide numerous other facilities and services to operators and emergency teams, by conducting behavioral analysis on incoming video feeds and detecting unusual or suspicious behavior. Behavioral analysis itself can be applied to numerous features extracted from video sequences including path detection and classification of which several methods are reviewed here. In this paper, we investigated a fuzzy inference engine approach to identify the human trajectories based on the paths that had been modeled by a self-learning system.

Keywords: *Intelligent video analysis, path detection, video content analytics, behavioral analysis.*

1. INTRODUCTION

Many surveillance systems today provide only a passive form of site monitoring. Extensive video records may be

kept to help find the instigator of criminal activities after the crime has been committed but preventive measures require human involvement. In addition, there is a need for large amounts of data storage to keep up to several large volumes of video streams that may be needed for later analysis. However, monitoring and storage space are not the only concerns. Even if these costs can be borne, there is the additional problem of reviewing this vast amount of video data AFTER an incident has occurred.

The goal of this work is to investigate one aspect of an intelligent video surveillance system that can help the relevant authorities speed up response times by having automatic, real-time alert mechanisms in place when suspicious behavior is detected. Behavioral analysis itself can be applied to numerous features extracted from video sequences including path detection and other aspects of human behaviour. Up till now, path classification has been carried out mainly using Boolean logic and allows only the identification of unusual paths, and not the extent to which they are deviant from usual paths. This paper reports on the results to solve this problem with a fuzzy inference approach to classify paths into different categories. The motivation for this comes from the fact that numerous fuzzy classification and clustering algorithms have already been used and proved to be efficient in several image processing and feature classification at various levels of abstraction in the image ranging from pixel-level to feature-level [1] [2]. Fuzzy-based inference has been proven to be advantageous in providing a 'human' judgment due to its ability to handle less rigid rules as well as the overlapping classification sets.

2. PREVIOUS WORK

Boyd et al [3] used an approach from computer network modeling (called network tomography) to study the flow of blobs in an image. They split the original image (figure 1a) up into several smaller cells (figure 1b) and recorded the number of entry and exits from once cell into all its adjacent cells. In this way they were able to produce a traffic intensity network model of the scene and identify possible areas in the region which served as *sources* and *sinks* for object trajectories. Coupled with the accumulated

statistics, this effectively generated a map of the region shown in figure 1(c):

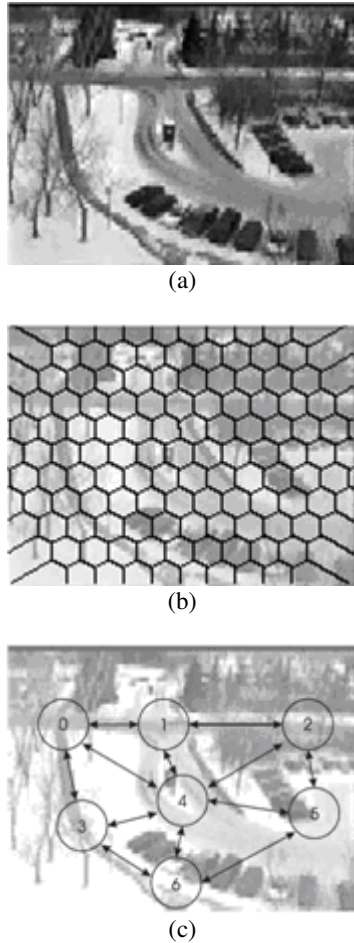


Figure 1 : Network tomography to study trajectories

Although this is quite an effective technique used in path analysis, it offers poor path extraction for video surveillance. Firstly, there is no real ‘path’ which is constructed. There is simply a mean count of flow between adjacent cells. This does not give a mathematically descriptive view of the average paths. Secondly, the extracted information, though useful in determining mean behavior, does not facilitate any comparison of a new trajectory with old ones, i.e. it will not serve any function for path behavioral analysis.

Johnson and Hogg [4] approached the problem of path extraction by suggesting a vector quantization approach where they lay down a number of formalisms which are useful for mathematical analysis of path extraction. They are among the first to lay down the proper notation and theoretical basis for several aspects of the problem, including a clear definition of trajectory (1) and flow (2):

$$T_i = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_{n-2}, y_{n-2}), (x_{n-1}, y_{n-1}), (x_n, y_n)\} \quad (1)$$

$$f = (x, y, \delta x, \delta y) \quad (2)$$

In order to quantize the vectors, Johnson and Hogg used two neural networks which are competitively engaged in unsupervised learning. Each node of the neural network corresponds to one randomly placed possible ‘prototype’ vector in the feature space. An object feature vector was then input into the network, and the node with the closest prototype was chosen based on the Euclidean distance. The chosen prototype was then updated by summing the original prototype as well as the difference between the input vector and the prototype multiplied by a ‘learning factor’. If a prototype is not the closest one to an incoming trajectory then it is not updated (3):

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \quad (3a)$$

$$m_i(t+1) = m_i(t) \text{ for } i \neq c \quad (3b)$$

However, Johnson and Hogg’s approach suffers from the same drawback of Boyd et al’s method: there is no mathematical representation of the path which can be used for comparison of new trajectory data. As such, there is limited applicability of this type of trajectory modeling in a behavioral analysis system. Zahn et al [5] also calculate a probability density function (PDF) of frequently used paths in a crowded scene. However, this is done without the use of a tracker. Their novel approach involves distinguishing background and foreground objects in the scene. An occurrence PDF which represents the density of foreground objects in different segments of the scene is calculated; and an orientation PDF which represents the flow of objects from one cell in the scene to another is also derived. By combining these two functions, they are able to segment the scene into regions of high occurrence and flow likelihood that are represented by spline interpolation. Makris and Ellis [6] use a spatial technique of modeling frequently used paths. They clearly describe a spatial method of defining paths as types of trajectories themselves with nodes, path boundaries, and node weights. Makris defined the distance measure used when comparing a trajectory with a path in order to measure the similarity between them. They do not use a simple Euclidean distance measure, because each path also has a route envelope which must be considered. The distance between the left and right closest boundary is also considered in the defined distance measure. Makris et al’s work is important in its abstraction and definition of a path. Their representation of paths is suitable for comparison with new trajectories to determine anomalies, although they themselves ‘have not yet considered how that model will be used to identify typical motions[6]. Currently, their model only takes into account the spatial layout of the path. Nevertheless a main drawback of their method is the arbitrarily defined distance measure. Euclidean metrics have been avoided as the model is not one single straight line, but rather a central path line with boundaries on either side of it.

3. SYSTEM OVERVIEW

Intelligent video surveillance for path analysis comprises of a number of key components as illustrated in figure 2.

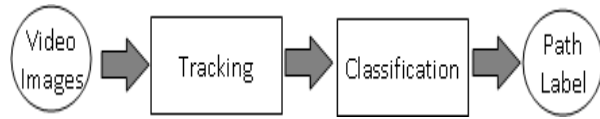


Figure 2 : Data flow for path analysis

The first task is to accurately track the object(s) of interest. Once we have the trajectories, we may then identify the paths. As the main focus of this work is to analyse the paths, pre-processing of the video data was done with the open source tracker *SwisTrack* [7]. It was developed by the *Distributed Intelligent Systems and Algorithms Laboratory* (DISAL) and the LPM Vision Group at EPFL, Lausanne, Switzerland and uses Intel's *OpenCV* library for fast image processing. Simple manual tracking was then done on the pre-processed results.

Path extraction or path detection refers to the grouping of similar object trajectories in video sequences and creating a path during the training phase. A set of similar trajectories are grouped together to form a path, bounded by some predefined tolerance level, to make up the path envelope. Such a model, attributed to Makris and Ellis [5] is illustrated in figure 3. The work here uses the spine of the path envelope as a reference for the normal path.

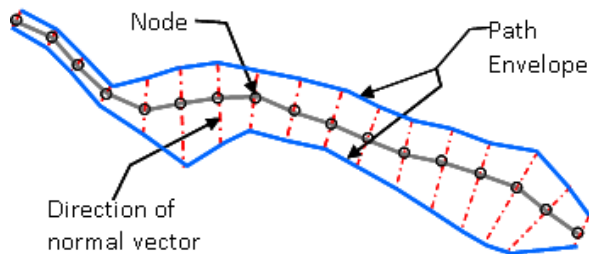


Figure 3 :Path Model

Intuitively, a path will represent an average of a set of similar trajectories. The extracted paths for each location would be stored in the database during the path extraction phase. These would then be used by the path classifier to evaluate all new trajectories during the testing phase. The overall path classification system is shown in figure 4.

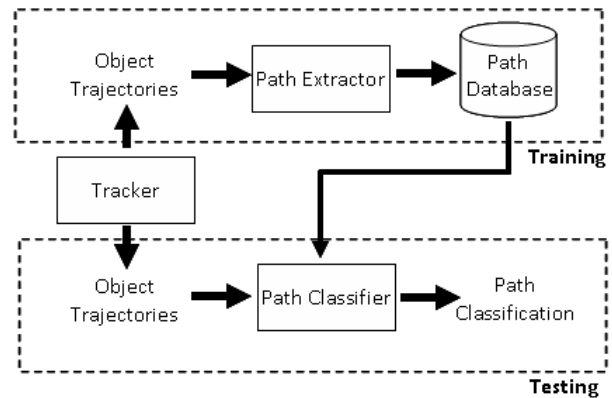


Figure 4 Path classification

The fuzzy inference engine developed would take two key inputs, as shown in figure 5. The first is the commonly used paths that are found in the scene which represents 'typical' or normal behavior. The other is any new object trajectories that the tracker extracts from a video sequence. The trajectory is then compared to the extracted paths on a multi-dimensional feature space. The set of features used are discussed in the next section.

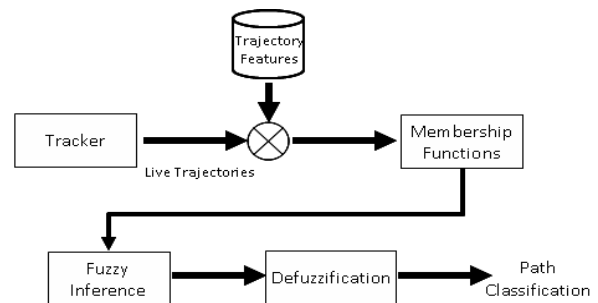


Figure 5 : Fuzzy classifier

4. RESULTS & DISCUSSION

The paths are modeled as a set of multidimensional features, viz.

- a) *RMS_Error* : Measures the error between the true path and the current (new) path.
- b) *Distance_Diff* : The difference in the total length of the whole path.
- c) *Speed_Diff* : The difference in the average speeds of the two trajectories.
- d) *Count_Diff* : The absolute difference in the number of discrete samples.

Each feature would have a range of values that would represent 'identical' to 'completely different'. These would then be passed through a set of fuzzy membership functions to get membership values corresponding to LOW, MEDIUM or HIGH, after which it would be finally passed through the set of IF-THEN rules. Some of these rules are shown in table 1.

1	If RMS_error is LOW and distance_diff is LOW and speed_diff is LOW and count_diff is LOW then <u>very usual</u>
2	If RMS_error is LOW and distance_diff is LOW and speed_diff is LOW and count_diff is MEDIUM then <u>very usual</u>
3	If RMS_error is LOW and distance_diff is LOW and speed_diff is MEDIUM and count_diff is LOW then <u>very usual</u>
4	If RMS_error is LOW and distance_diff is LOW and speed_diff is MEDIUM and count_diff is MEDIUM then <u>very usual</u>
5	If RMS_error is LOW and distance_diff is MED and speed_diff is LOW and count_diff is LOW then <u>usual</u>
6	If RMS_error is LOW and distance_diff is MED and speed_diff is LOW and count_diff is MEDIUM then <u>usual</u>
7	If RMS_error is LOW and distance_diff is MED and speed_diff is MEDIUM and count_diff is LOW then <u>usual</u>
8	If RMS_error is LOW and distance_diff is MED and speed_diff is MEDIUM and count_diff is MEDIUM then <u>usual</u>
...	...
...	...
...	...
...	...
30	If RMS_error is HIGH then <u>very suspicious</u>
31	If RMS_error is HIGH and distance_diff is HIGH then <u>very suspicious</u>
32	If RMS_error is HIGH OR distance_diff is HIGH and speed_diff is HIGH then <u>very suspicious</u>

Table 1: The Fuzzy IF-THEN rules

The fuzzy inference engine passes them through these 32 rules where each of the rules would generate a response. The system would then defuzzify the firing strengths for all the responses, corresponding to ‘Very Usual’, ‘Usual’, ‘Usual or Suspicious’, ‘Suspicious’ and ‘Very Suspicious’ to obtain the crisp output which would be a single value that ranges from 0 to a maximum of 100. Defuzzification of the responses is done with the Centroid method, which basically returns the centre of the mass for the shape of the output curve. The output membership function corresponding to each of these 5 responses is shown in figure 6.

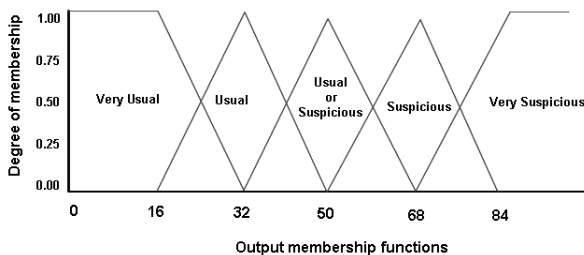


Figure 6: Fuzzy membership functions

The final step involves thresholding the crisp output to produce an alert for abnormal behavior if this is less than

50. Extracted paths from 18 different scenarios based on two locations were collected and analysed. Figure 7 shows these two locations and the extracted normal paths.

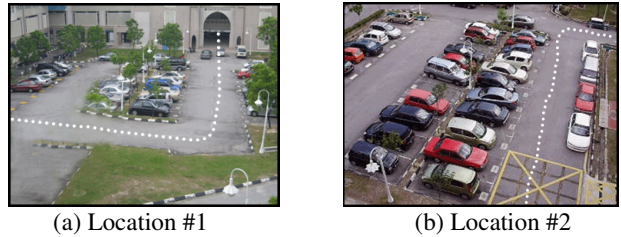


Figure 7 : Extracted normal paths for 2 locations

Abnormal paths were also collected. Figure 8 shows two examples of the new paths which were tested against the normal paths.



(a) Variation of normal path



(b) Highly abnormal path

Figure 8: Test Data

Finally, figure 9 shows the screenshot of the path analysis system with the fuzzy-based path classifier.

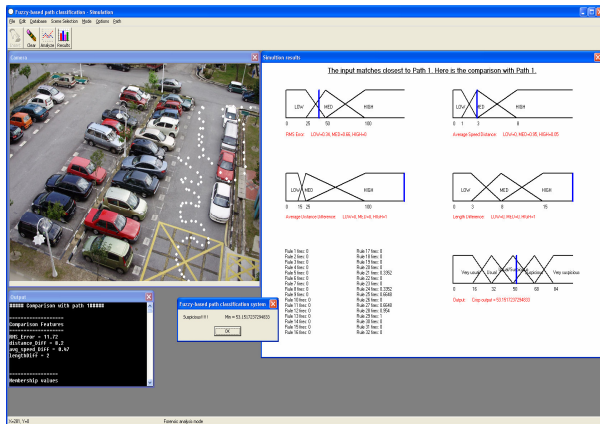


Figure 9: System output

Moreover, a Boolean logic system and a neural network (MLP) were also tested to compare the performance. The neural network used here consists of 1 hidden layer with 6 neurons and trained with the standard back propagation algorithm, with 30 examples in 1,000 epochs. 39 human subjects were asked to validate whether each path is suspicious/abnormal for the extracted paths in each of these 18 different scenarios. Their responses were then compared with those obtained from the three system, viz. Fuzzy, Boolean and Neural approaches. The Boolean approach computes the linear weighted sum of the feature set.

$$\text{Output} = (\alpha_1 \text{DistanceDiff1} + \alpha_2 \text{SpeedDiff} + \alpha_3 \text{CountDiff} + \alpha_4 \text{RMS}) / 100$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are weighted coefficients.

These coefficients are then tuned by supervised learning. The following table summarizes the results.

	Fuzzy logic	Boolean logic	Neural Network
Humans validation	12	9	11
Accuracy	68%	50%	61%

Table 2 : Results verified by humans

5. CONCLUSIONS

This paper has shown the practicality of using a fuzzy-based path classification to analyse the trajectory data extracted from image sequences. The fuzzy system tends to be ‘pessimistic’ in its inaccuracy – i.e. finding more paths suspicious, rather than failing to detect suspicious paths. One approach to solve this is to alter the membership functions to increase tolerance levels – if the goal is to get the system to agree with the human’s analysis. Nevertheless, we believe that such a fuzzy-based path classification system can provide additional inputs to strengthen the security of the location that is under surveillance. Adding a degree of intelligence to these video surveillance systems does minimize some of the

difficulty attributed to data overload when one has too many scenes to monitor, by automatically alerting the operator of certain events.

Higher-level classification categories can also be built using a fuzzy approach and the basic metrics used here. Output classes for high, medium and low degrees of tailing, loitering, etc can also be used for classification. As the current work is limited to a single camera and scene, the next logical step of this piece of investigation is to extend the model to cater for paths that may extend over two cameras.

6. REFERENCES

1. S. Sinha, F. Karray, P. Fieguth “Underground Pipe Cracks Classification Using Image Analysis and Neuro-Fuzzy Algorithm”. Proceedings of the IEEE International Symposium on Intelligent Control and Intelligent Systems and Semiotics, Cambridge, MA September 15-17, 1999.
2. I. Nedeljkovic “Image Classification Based On Fuzzy Logic. XXth ISPRS Congress, International Society for Photogrammetry and Remote Sensing, 12-23 July 2004 Istanbul, Turkey, Commission 6.
3. J. E. Boyd, J. Meloche, and Y. Vardi, “Statistical Tracking in Video Traffic Surveillance”, in Proc. ICCV99, Corfu Greece, Sept. 1999.
4. N. Johnson and D. Hogg, “Learning the Distribution of Object Trajectories for Event Recognition”, in Proc. BMVC95, Birmingham, England, 1995.
5. B. Zahn, P Remagnino and S.A. Velastin, “Mining Paths of Complex Crowd Scenes”. In *Lecture Notes in Computer Science*, vol. 3804, pp 126-133 (G Bebis et al, eds.), Springer-Verlag, Berlin, 2005
6. D. Makris and T. Ellis, "Path Detection in Video Surveillance", *Image and Vision Computing Journal*, vol. 20/12, pp 895-903, October 2002
7. T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, “SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems”, Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS 2008), pages 4004-4010, 2008.